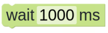
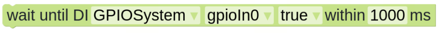
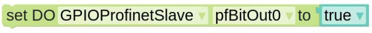
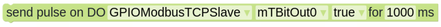
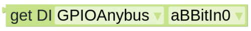
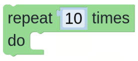
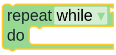
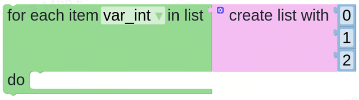
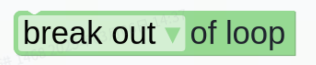
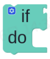
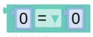
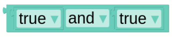
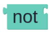
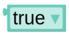
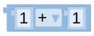
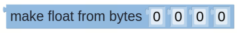
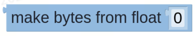
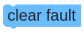
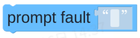
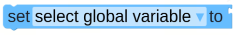
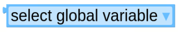
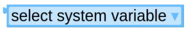
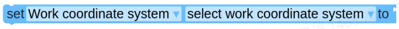
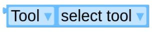
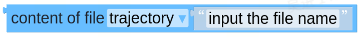
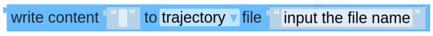
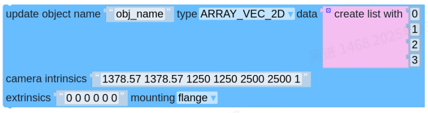
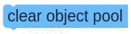
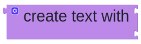
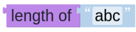
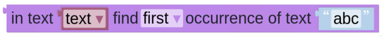
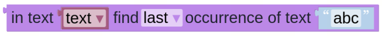
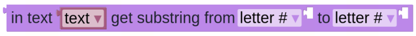
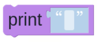
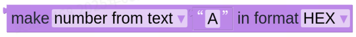
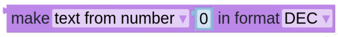
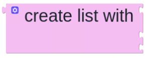
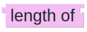
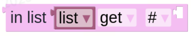
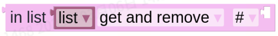
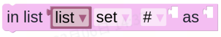
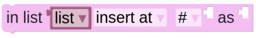
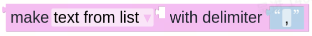
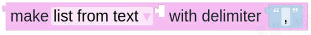
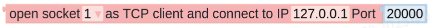
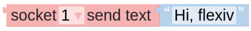
# Blockly VS Python

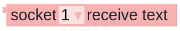
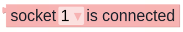
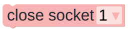
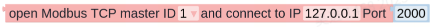
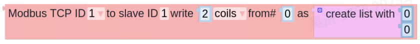
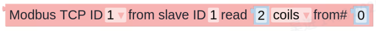
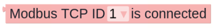
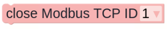
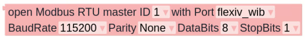
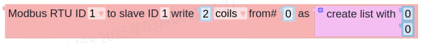
| | Blockly | Python |
|---|---|---|
| Signal | <br>wait 1000 ms | `wait_ms(1000)` |
| | <br>wait until DI GPIOSystem gpioIn0 true within 1000 ms | `wait_io_ms("GPIOSystem", "gpioIn0", True, 1000)` |
| | <br>set DO GPIOProfinetSlave pfBitOut0 to true | `set_io("GPIOProfinetSlave", "pfBitOut0", True)` |
| | <br>send pulse on DO GPIOModbusTCPSlave mTBitOut0 true for 1000 ms | `set_io_pulse_ms("GPIOModbusTCPSlave", "mTBitOut0", True, 1000)` |
| | <br>get DI GPIOAnybus aBBitIn0 | `get_io("GPIOAnybus", "aBBitIn0")` |
| Loops | <br>repeat 10 times<br>do | `for i in range(10):`<br>`    pass` |
| | <br>repeat while<br>do | `while False:`<br>`    pass` |
| | <br>for each item var_int in list create list with 0 1 2<br>do | `for var_int in [1, 2, 3]:`<br>`  pass` |

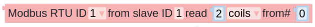
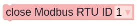
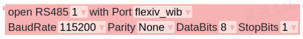
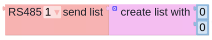
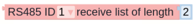
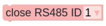
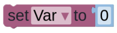
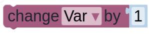
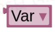
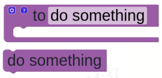
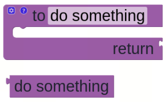
|  |  |  |
|---|---|---|
|  |  | `break` |
| Logic |  | `if False:`<br>`    pass` |
|  |  | `0 == 0` |
|  |  | `True and True` |
|  |  | `not` |
|  |  | `True` |
| Math | | `0` |
|  |  | `1 + 1` |
|  |  | `from struct import unpack`<br>`unpack('f', b'\x00\x00\x00\x00')[0]` |
|  |  | `from struct import pack`<br>`pack('f', 0)` |

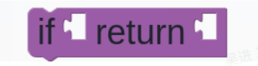
|  |  |  |
|---|---|---|
| Robot |  clear fault | `clear_fault()` |
|  |  prompt fault " " | `fault()` |
|  |  set select global variable ▾ to | `set_global_var(var_name, value)` |
|  |  select global variable ▾ | `get_global_var(var_name)` |
|  |  select system variable ▾ | `get_system_state(system_var)` |
|  |  set Work coordinate system ▾ select work coordinate system ▾ to | `set_workcoord_or_tool("Work coordinate system", workcoord_name)` |
|  |  Tool ▾ select tool ▾ | `get_workcoord_or_tool("Tool", tool_name)` |
|  |  content of file trajectory ▾ " input the file name " | `content_of_file("trajectory", traj_file_name)` |
|  |  write content " " to trajectory ▾ file " input the file name " | `write_trajectory_file(content, traj_file_name)` |
|  |  | `update_object_pool("obj_name",` |

|  |  |  |
|---|---|---|
|  | <br>update object name obj_name type ARRAY_VEC_2D data create list with 0 1 2 3<br>camera intrinsics 1378.57 1378.57 1250 1250 2500 2500 1<br>extrinsics 0 0 0 0 0 0 mounting flange | `25, [0, 1, 2, 3], "1378.57 1378.57 1250 1250 2500 2500 1", "0 0 0 0 0 0",`<br>`"flange")` |
|  | <br>clear object pool | `clear_object_pool()` |
| **Text** | <br>“ ” | `""` |
|  | <br>create text with | `concat_string("", "")` |
|  | <br>length of “ abc ” | `len("abc")` |
|  | <br>in text text find first occurrence of text “ abc ” | `first_index_string(text, "abc")` |
|  | <br>in text text find last occurrence of text “ abc ” | `last_index_string(text, "abc")` |
|  | <br>in text text get substring from letter # to letter # | `sub_string(text, start_index, end_index)` |
|  | <br>print “ ” | `print("")` |
|  | <br>make number from text “ A ” in format HEX | `to_number("A", 16)` |
|  |  |  |

| | | |
|---|---|---|
| |  make text from number 0 in format DEC | `to_string("0", 10)` |
| List |  create list with | `[None, None, None]` |
| |  length of | `len(list)` |
| |  in list list get # | `get_list(list, index)` |
| |  in list list get and remove # | `remove_list(list, index)` |
| |  in list list set # as | `set_list(list, index, value)` |
| |  in list list insert at # as | `insert_list(list, index, value)` |
| |  make text from list with delimiter “ , ” | `join_string(list, delimiter)` |
| |  make list from text with delimiter “ , ” | `split_string(text, delimiter)` |
| Communication |  open socket 1 as TCP client and connect to IP 127.0.0.1 Port 20000 | `socket_open(1, "127.0.0.1", 20000)` |
| |  socket 1 send text “ Hi, flexiv ” | `socket_send(1, "Hi, flexiv")` |

| | | |
|---|---|---|
| | <br>socket 1 receive text | `socket_recv(1)` |
| | <br>socket 1 is connected | `socket_connected(1)` |
| | <br>close socket 1 | `socket_close(1)` |
| | <br>open Modbus TCP master ID 1 and connect to IP 127.0.0.1 Port 2000 | `modbus_tcp_open(1, "127.0.0.1", 2000)` |
| | <br>Modbus TCP ID 1 to slave ID 1 write 2 coils from# 0 as create list with 0 0 | `modbus_tcp_write(1, 1, 2, 0, 0, [0, 0])` |
| | <br>Modbus TCP ID 1 from slave ID 1 read 2 coils from# 0 | `modbus_tcp_read(1, 1, 2, 0, 0)` |
| | <br>Modbus TCP ID 1 is connected | `modbus_tcp_connected(1)` |
| | <br>close Modbus TCP ID 1 | `modbus_tcp_close(1)` |
| | <br>open Modbus RTU master ID 1 with Port flexiv_wib<br>BaudRate 115200 Parity None DataBits 8 StopBits 1 | `modbus_rtu_open(1, 'flexiv_wib', 115200, 0, 8, 1)` |
| | <br>Modbus RTU ID 1 to slave ID 1 write 2 coils from# 0 as create list with 0 0 | `modbus_rtu_write(1, 1, 2, 0, 0, [0, 0])` |

| | | |
|---|---|---|
| |  Modbus RTU ID 1 from slave ID 1 read 2 coils from# 0 | `modbus_rtu_read(1, 1, 2, 0, 0)` |
| |  close Modbus RTU ID 1 | `modbus_rtu_close(1)` |
| |  open RS485 1 with Port flexiv_wib<br>BaudRate 115200 Parity None DataBits 8 StopBits 1 | `serial_port_open(1, 'flexiv_wib', 115200, 0, 8, 1)` |
| |  RS485 1 send list create list with 0 0 | `serial_port_send(1,({0, 0}))` |
| |  RS485 ID 1 receive list of length 2 | `serial_port_recv(1,2)` |
| |  close RS485 ID 1 | `serial_port_close(1)` |
| Variables |  set Var to 0 | `var = 0` |
| |  change Var by 1 | `var = var + 1` |
| |  Var | `var` |
| Functions |  to do something<br>do something | |
| |  to do something<br>return<br>do something | |

|  |  |  |  |
|---|---|---|---|
|  | if<br>return |  |  |